IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Patent Application of
GILKERSON
Serial No. 10/779,808
Filed: February 18, 2004

Confirmation No.: 6826
Atty. Ref.: 550-525
Group: 2183
Examiner: R. Fennema

For: DETERMINING TARGET ADDRESSES FOR INSTRUCTION FLOW CHANGING INSTRUCTIONS IN A DATA PROCESSING APPARATUS

* *

APPEAL BRIEF

On Appeal From Group Art Unit 2183

Stanley C. Spooner
NIXON & VANDERHYE P.C.
11th Floor, 901 North Glebe Road
Arlington, Virginia 22203
(703) 816-4028
Attorney for Appellant

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST 1

II. RELATED APPEALS AND INTERFERENCES 2

III. STATUS OF CLAIMS 2

IV. STATUS OF AMENDMENTS 3

V. SUMMARY OF THE CLAIMED SUBJECT MATTER 3

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL 10

VII. ARGUMENT 10

A. The Furber prior art, where there is an instruction flow changing instruction [IFCI], does not teach a determination between two address generation paths based upon the IFCI being a first prefetched instruction 11

B. The Furber reference does not teach "the first address generation path generating the target address more quickly than the at least one further address generation path" 14

C. The Patterson reference is not alleged to disclose the above two features clearly missing from the Furber reference 16

D. The combination of the Furber and Patterson references fails to teach the claimed invention 17

E. The Examiner fails to identify any "reason" or "motivation" for combining the Furber and Patterson references 17

F. The Examiner fails to appreciate that Furber would lead those of ordinary skill in the art away from the claimed invention 19

G. Claims 1-6, 10-16, 20 and 21 are not obvious under 35 USC §103 over Furber/Patterson 19

H. Claims 7-9 and 17-19 are not obvious under 35 USC §103 over the Furber/Patterson/Hara 21

VIII. CONCLUSION 22

IX. CLAIMS APPENDIX 1

X. EVIDENCE APPENDIX 11

XI. RELATED PROCEEDINGS APPENDIX 12

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Patent Application of

GILKERSON — Atty. Ref.: 550-525

Serial No. 10/779,808 — Group: 2183

Filed: February 18, 2004 — Examiner: R. Fennema

For: DETERMINING TARGET ADDRESSES FOR INSTRUCTION FLOW CHANGING INSTRUCTIONS IN A DATA PROCESSING APPARATUS

* *

May 9, 2007

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

I. REAL PARTY IN INTEREST

The real party in interest in the above-identified appeal is ARM Limited by virtue of an assignment of rights from the inventor to ARM Limited recorded July 12, 2004 at Reel 15561, Frame 815.

II. RELATED APPEALS AND INTERFERENCES

There are believed to be no related appeals, interferences or judicial proceedings with respect to the present application, other than the Pre-Appeal Brief Request for Review previously filed in this appeal.

III. STATUS OF CLAIMS

Claims 1-21 stand rejected in the Final Official Action. The Examiner contends that claims 1-6, 10-16, 20 and 21 are obvious under 35 USC §103 over Furber ("ARM System-on-Chip Architecture" published 2000) in view of Patterson (presumably a reference to "Computer Architecture – A Quantitative Approach" authored by Hennessy and Patterson). The Examiner also contends that claims 7-9 and 17-19 are obvious under 35 USC §103 over the Furber/Patterson combination, further in view of Hara (U.S. Patent 5,848,269). With respect to both rejections under §103, the Examiner contends that the combined references disclose each and every element and interrelationship between elements as set out in Appellant's claims and, in addition, that there is some reason or motivation for combining these references. The above rejections of claims 1-21 are appealed.

IV. STATUS OF AMENDMENTS

No further response has been submitted with respect to the Final Official Action in this application other than the filing of a Pre-Appeal Brief Request for Review on January 26, 2007 and the decision mailed on March 9, 2007.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

Appellant's specification and figures provide an explanation of the claimed invention set out in independent claims 1, 11 and 21, with each claimed structure addressed as to its location in the specification and in the figures.

"1. A data processing apparatus, comprising:

a processor [item 30 in fig. 1 and discussed on page 10, lines 7-13 and elsewhere in the specification] operable to execute a stream of instructions;

a prefetch unit [item 20 in fig. 1 and discussed on page 10, lines 7-13 and elsewhere in the specification] operable to prefetch instructions from a memory [item 10 in fig. 1 and discussed on page 10, lines 7-13 and elsewhere in the specification] prior to sending those instructions to the processor for execution, the prefetch unit [20] being operable to receive from the memory [10] simultaneously a plurality of prefetched instructions from sequential addresses in memory, and being operable to detect [by way of decode logic 70 and prefetch control unit 50 shown in fig. 1 and discussed on page 11, line 17 to page 13, line 23 and elsewhere in the specification]

whether any of those prefetched instructions are an instruction flow changing instruction, and based thereon to output a fetch address [from register 48 shown in fig. 1 and discussed on page 12, lines 13-23 and elsewhere in the specification] for a next instruction to be prefetched by the prefetch unit [20];

address generation logic, within the prefetch unit and responsive to a selected prefetched instruction that is detected to be said instruction flow changing instruction, [decode logic 70, adder logic 90, register 105, increment logic 110 and multiplexer 115, all shown in fig. 1 and discussed on page 11, line 17 – page 12, line 7 and page 13, line 24 to page 14, line 5 and elsewhere in the specification] for determining a target address to be output as the fetch address, the address generation logic having a first address generation path [through decode block 78, multiplexer 85, adder logic 90, multiplexer 46 as shown in fig. 1 and discussed on page 14, line 29 to page 15, line 3 and elsewhere in the specification] for determining the target address if the selected prefetched instruction is a first prefetched instruction in said plurality, and at least one further address generation path [through decode blocks 72/74/76, offset multiplexer 80, offset register 82, multiplexer 85, adder logic 90, multiplexer 46 as shown in fig. 1 and discussed on page 14, line 29 to page 15, line 3 and elsewhere in the specification] for determining the target address if the selected prefetched instruction is one of the other prefetched instructions in said plurality, the first prefetched instruction being earlier in said stream than said other prefetched instructions [holding register 95 in fig. 1 and discussed on page 12, line 24 to page

13, line 2 and elsewhere in the specification], the first address generation path generating the target address more quickly than the at least one further address generation path [discussed on page 14, lines 15-28 and elsewhere in the specification]; and

a pipeline stage [offset register 82 and multiplexer 80 as shown in fig. 1 and described on page 15, lines 3-7 and elsewhere in the specification], provided in said at least one further address generation path, for increasing generation speed of the target address by the first address generation path, whereby, in the event that the first prefetched instruction is said selected prefetched instruction, the prefetch unit outputs the associated target address as the fetch address earlier than if one of said other prefetched instructions is said selected prefetched instruction [discussed on page 14, lines 15-28 and elsewhere in the specification]."

"11. A method of operating a data processing apparatus to determine a target address for an instruction flow changing instruction, the data processing apparatus having a processor [item 30 in fig. 1 and discussed on page 10, lines 7-13 and elsewhere in the specification] operable to execute a stream of instructions, and a prefetch unit [item 20 in fig. 1 and discussed on page 10, lines 7-13 and elsewhere in the specification] operable to prefetch instructions from a memory [item 10 in fig. 1 and discussed on page 10, lines 7-13 and elsewhere in the specification] prior to sending those instructions to the processor for execution, and to output a fetch

address [from register 48 shown in fig. 1 and discussed on page 12, lines 13-23 and elsewhere in the specification] for a next instruction to be prefetched from the memory [10], the method comprising the steps of:

(a) receiving from the memory [10] simultaneously a plurality of prefetched instructions (holding register 95 shown in fig. 1 and discussed on page 12, line 17 to page 13, line 10 and elsewhere in the specification] from sequential addresses in memory;

(b) detecting whether any of those prefetched instructions are an instruction flow changing instruction [decode logic 70 and prefetch control unit 50 shown in fig. 1 and discussed on page 13, lines 2-10 and elsewhere in the specification]; and

(c) for a selected prefetched instruction that is detected to be said instruction flow changing instruction, determining a target address to be output as the fetch address [from register 48 shown in fig. 1 and discussed on page 12, lines 13-23 and elsewhere in the specification] by performing one of the steps of:

(c)(1) employing a first address generation path [through decode block 78, multiplexer 85, adder logic 90, multiplexer 46 as shown in fig. 1 and discussed on page 14, line 29 to page 15, line 3 and elsewhere in the specification] to determine the target address if the selected prefetched instruction is a first prefetched instruction in said plurality; or

(c)(2) employing at least one further address generation path [through decode blocks 72/74/76, offset multiplexer 80, offset register 82, multiplexer 85, adder logic 90, multiplexer 46 as shown in fig. 1 and discussed on page 14, line 29 to page 15, line 3 and elsewhere in the specification] to determine the target address if the selected prefetched instruction is one of the other prefetched instructions in said plurality;

the first prefetched instruction being earlier in said stream than said other prefetched instructions [holding register 95 in fig. 1 and discussed on page 12, line 24 to page 13, line 2 and elsewhere in the specification], and the first address generation path being arranged to generate the target address more quickly than the at least one further address generation path [discussed on page 13, lines 2-10 and elsewhere in the specification];

(d) providing a pipeline stage [offset register 82 and multiplexer 80 as shown in fig. 1 and described on page 15, lines 3-7 and elsewhere in the specification] in the at least one further address generation path in order to increase speed of generation of the target address by the first address generation path; and

(e) outputting as the fetch address the target address generated at step (c) [address adder 90 shown in fig. 1 and discussed on page 14, lines 2-7 and elsewhere in the specification];

whereby, in the event that the first prefetched instruction is said selected prefetched instruction, the prefetch unit is operable to output the associated

target address as the fetch address earlier than if one of said other prefetched instructions is said selected prefetched instruction [discussed on page 14, lines 15-28 and elsewhere in the specification]."

"21. A prefetch unit [item 20 in fig. 1 and discussed on page 10, lines 7-13 and elsewhere in the specification] for a data processing apparatus that has a processor [item 30 in fig. 1 and discussed on page 10, lines 7-13 and elsewhere in the specification] operable to execute a stream of instructions, the prefetch unit [20] being operable to prefetch instructions from a memory [item 10 in fig. 1 and discussed on page 10, lines 7-13 and elsewhere in the specification] prior to sending those instructions to the processor [20] for execution, the prefetch unit being operable to receive from the memory [10] simultaneously a plurality of prefetched instructions from sequential addresses in memory, and being operable to detect whether any of those prefetched instructions are an instruction flow changing instruction, and based thereon to output a fetch address [address adder 90 shown in fig. 1 and discussed on page 14, lines 2-7 and elsewhere in the specification] for a next instruction to be prefetched by the prefetch unit, the prefetch unit comprising:

address generation logic, responsive to a selected prefetched instruction that is detected to be said instruction flow changing instruction [decode logic 70, adder logic 90, register 105, increment logic 110 and multiplexer 115, all shown in fig. 1 and discussed on page 11, line 17 – page 12, line 7 and page 13, line 24 to page 14,

line 5 and elsewhere in the specification], for determining a target address to be output as the fetch address, the address generation logic having a first address generation path [through decode block 78, multiplexer 85, adder logic 90, multiplexer 46 as shown in fig. 1 and discussed on page 14, line 29 to page 15, line 3 and elsewhere in the specification] for determining the target address if the selected prefetched instruction is a first prefetched instruction in said plurality, and at least one further address generation path [through decode blocks 72/74/76, offset multiplexer 80, offset register 82, multiplexer 85, adder logic 90, multiplexer 46 as shown in fig. 1 and discussed on page 14, line 29 to page 15, line 3 and elsewhere in the specification] for determining the target address if the selected prefetched instruction is one of the other prefetched instructions in said plurality, the first prefetched instruction being earlier in said stream than said other prefetched instructions [holding register 95 in fig. 1 and discussed on page 12, line 24 to page 13, line 2 and elsewhere in the specification], the first address generation path generating the target address more quickly than the at least one further address generation path [discussed on page 14, lines 15-28 and elsewhere in the specification]; and

a pipeline stage [offset register 82 and multiplexer 80 as shown in fig. 1 and described on page 15, lines 3-7 and elsewhere in the specification], provided in said at least one further address generation path, for increasing generation speed of the target address by the first address generation path, whereby, in the event that the

first prefetched instruction is said selected prefetched instruction, the prefetch unit outputs the associated target address as the fetch address earlier than if one of said other prefetched instructions is said selected prefetched instruction [discussed on page 14, lines 15-28 and elsewhere in the specification]."

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-6, 10-16, 20 and 21 stand rejected under 35 USC §103 as being unpatentable over Furber in view of Patterson.

Claims 7-9 and 17-19 stand rejected under 35 USC §103 as being unpatentable over the Furber/Patterson combination, further in view of Hara.

VII. ARGUMENT

Appellant's arguments include the fact that the burden is on the Examiner to first and foremost properly construe the language of the claims to determine what structure and/or method steps are covered by that claim. After proper construction of the claim language, the burden is also on the Examiner to demonstrate where a single reference (in the case of anticipation) or a plurality of references (in the case of an obviousness rejection) teaches each of the structures and/or method steps recited in independent claims 1, 11 and 21

Furthermore, the Court of Appeals for the Federal Circuit has stated in the case of *In re Rouffet*, 47 USPQ2d 1453, 1458 (Fed. Cir. 1998)

> to prevent the use of hindsight based on the invention to defeat patentability of the invention, this court **requires** the examiner to show a **motivation** to combine the references that create the case of obviousness. In other words, the Examiner **must show reasons** that the skilled artisan, confronted with the same problems as the inventor and with no knowledge of the claimed invention, would select the elements from the cited prior art references for combination in the manner claimed. (Emphasis added).

A. The Furber prior art, where there is an instruction flow changing instruction [IFCI], does not teach a determination between two address generation paths based upon the IFCI being a first prefetched instruction

Appellant's independent claims 1, 11 and 21 recite address generation logic within the prefetch unit "having a first address generation path for determining the target address **if** the selected prefetched instruction is a first prefetched instruction in said plurality, **and** at least one further address generation path for determining the target address **if** the selected prefetched instruction is one of the other prefetched instructions in said plurality." (emphasis added). The Examiner alleges that the claimed apparatus and method is disclosed at page 388 of the Furber reference.

The Furber reference is a document written and published by the assignee of the Appellant's application and is well know to the Appellant. The only portion of the Furber reference relating to the above quoted portion of Appellant's claim is the "instruction prefetch unit" which discussion starts at the bottom page 387 and goes

through the first six lines on page 388. Figure 14.10 in Furber is a figure of the overall "AMULET3 core organization." It should be noted that the "prefetch unit" is only the top block in that figure. From a review of this portion of Furber it will be seen that it discloses a branch prediction unit which is split into two 8-entry halves with branches at **even** half-word addresses **stored in one half** and branches at **odd** half-word addresses **stored in the other half**.

The Examiner suggests that these two halves of the branch prediction unit in Furber correspond to Appellant's "first" and "further address generation" paths. However, this suggestion contradicts the literal language of the independent claims which specify that, where a selected prefetched instruction is an instruction flow changing instruction (hereinafter "**IFCI**") a first address generation path determines the target address **if** one circumstance occurs and a further address generation path determines the target address **if** any other circumstance occurs, i.e., the use of the paths is asymetric.

Essentially, if there is at least one IFCI, the address generation logic is used to make the decision as to which of at least two address generation paths to follow. In claims 1, 11 and 21, the **first** address generation path is used "**if** the selected prefetched instruction [by definition an IFCI] is a first prefetched instruction," and the **at least one further** address generation path is used "**if** the selected prefetched instruction [by definition an IFCI] is one of the other prefetched instructions."

In contrast to the language of independent claims 1, 11 and 21, in Furber, **any** "even" half-word address is handled by the first path and **any** "odd" half-word address would be handled by the second path, i.e., the use of the paths is symmetric.

Moreover, there is simply no identification of the IFCI being first or not in Furber because Furber doesn't care. In Furber, the decision as to the address generation path is made based upon whether the instruction is at an odd or even half word address and has nothing to do with a recognition of the instruction being an IFCI and whether this IFCI is "first."

The Examiner is respectfully requested to identify in the Examiner's Answer where Furber teaches or suggests (1) any determination as to an instruction being an IFCI; and (2) determining whether the first instruction is an IFCI. Appellant believes the Examiner will be unable to make any identification of either of these claimed features, just as he has failed to identify these features in the previous Official Actions.

The Examiner's failure to consider the detailed "address generation logic" language in claims 1, 11 and 21 suggests that the Examiner has misinterpreted the teaching in the Furber reference. However, should the Examiner traverse this logic, he is again respectfully requested to identify where there are at least two separate address generation paths and, even more specifically, where there is any determination in Furber as to whether the selected instruction is an IFCI and if an IFCI whether it is "said first prefetched instruction." Absent such identification of

the claimed structure and claimed interrelationships (and method steps in the other independent claims) in the Furber patent by the Examiner in the Examiner's Answer, the rejection under 35 USC §103 clearly fails.

B. The Furber reference does not teach "the first address generation path generating the target address more quickly than the at least one further address generation path"

Appellant's independent claims 1, 11 and 21 also specifically require "the first address generation path generating the target address **more quickly** than the at least one further address generation path." (emphasis added). The Examiner also alleges that this is taught in Furber at page 388, which only states "the first path takes **priority**." (emphasis added).

The Examiner again apparently misunderstands the claim language which states "the first address generating path generating the target address **more quickly** than" the further path (emphasis added) with no mention about priority. The Examiner provides no evidence that "priority" is the same as "more quickly."

Moreover, there is no indication in Furber that any one of the two paths generates a target address "more quickly" than the other. In fact, Furber implies just the direct opposite, i.e., two 8-entry halves of the branch prediction instruction [IFCI] take the same time to operate. In fact Furber states that "a packet with a branch [IFCI] in each half word may 'hit' in both halves, whereupon the target of

the first instruction (at the even address) takes priority." Thus, because both paths take the same amount of time, Furber teaches that, when there is a simultaneous hit, the even address "takes priority." There is no mention of any speed advantage of one path over the other or one being generated "more quickly" than the other.

Appellant's claims are not directed towards **priority** of an instruction, i.e., which one to select, but rather, which instruction is generated more quickly and, with respect to the generation of target address for the selected prefetched instruction, there is no suggestion in Furber that target address generation occurs **more quickly** in one path than in another path.

The Examiner also admits that Furber fails to teach "a pipeline stage, provided in said one further address generation path, for increasing generation speed of the target address by the first address generation path." (Final Rejection, page 3). This admission supports the contention that Furber has nothing to do with using IFCI recognition to choose among flow paths or that one flow path operates more quickly than the others (if Furber doesn't teach any increasing of flow path speed, how can it suggest any decision logic as to which path to chose). The Examiner effectively admits that Furber does not disclose any difference in speed between the first and the at least one further flow paths.

Because Furber clearly fails to teach this feature which is positively recited in each of Appellant's independent claims 1, 11 and 21 (the "address generation logic" in the prefetch unit of claims 1 and 21 and the method steps in claim 11)

which feature is only the top block in Furber's Figure 14.10, there is simply no support for the Examiner's contention that this portion of Appellant's independent claims are rendered obvious in view of the Furber reference.

C. The Patterson reference is not alleged to disclose the above two features clearly missing from the Furber reference

The Examiner admission that Furber fails to teach "a pipeline stage, provided in said one further address generation path, for increasing generation speed of the target address by the first address generation path" (Final Rejection, page 3) implies that he relies upon Patterson to teach this claimed feature.

The Examiner alleges in his unsupported conclusory statement that Patterson teaches speeding up execution of an address path. However, he does not support this with any evidence of record tending to identify where Patterson generates any one target address "more quickly" than another target address. Should the Examiner believe otherwise, he is respectfully requested to cite the portion of Patterson relied upon for this claim teaching.

Additionally, the Examiner does not allege that Patterson teaches deciding between two paths based upon **if** the selected prefetched instruction [IFCI] is the "first prefetched instruction" or is "one of the other prefetched instructions." Again, should he believe this to be disclosed in Patterson, he is requested to

identify such teaching and, if unable to identify such teaching, admit that it doesn't exist.

D. The combination of the Furber and Patterson references fails to teach the claimed invention

Furber fails to teach (1) making a determination between two address generation paths based on an IFCI being the first prefetched instruction (in section A above) and (2) as admitted by the Examiner, fails to teach one path operating more quickly than another (in section B above). Given that the Examiner fails to identify any teaching in Patterson (in section C above) which supplies the teachings missing from the Furber reference, even if Furber and Patterson were combined, they cannot render obvious the subject matter of independent claims 1, 11 and 21.

E. The Examiner fails to identify any "reason" or "motivation" for combining the Furber and Patterson references

As noted above in section D, even if Furber and Patterson were combined, the combination would fail to disclose or render obvious the subject matter of Applicant's independent claims 1, 11 and 21. Moreover, such combination would not obvious unless there were some "reason" or "motivation" (see the Federal Circuit decision in *In re Rouffet*, 47 USPQ2d 1453, 1457-8 (Fed. Cir. 1988) as quoted above).

The Examiner glosses over this glaring problem by stating that Patterson's teaching of a pipeline processor and the possibility of a higher frequency clock cycle, somehow magically increases the speed of execution on only the first address path and slows down all other paths. There is no explanation by the Examiner. There is simply no identified suggestion or teaching in Patterson that this could or would be of any use in a prefetch unit, let alone in the address generation logic within the prefetch unit as set out in Applicant's independent claims.

The Examiner makes the conclusory statement that "it would have been obvious . . . to increase the clock rate for other paths and apply it to Furber's invention." (Final Rejection, section 4, page 4). The Examiner is simply applying circular reasoning, i.e., because of his error in ignoring the failure of Furber and Patterson to disclose all claimed elements and his imagined improvement provided by Patterson, that if Furber and Patterson were combined, they would be combined in such a fashion (as taught in the present specification) and provide the benefits of Appellant's claimed invention. This is simply not the test of obviousness and the Examiner has committed reversible error by ignoring the Federal Circuit's test of obviousness.

F. The Examiner fails to appreciate that Furber would lead those of ordinary skill in the art away from the claimed invention

The Court of Appeals for the Federal Circuit *In re Fine*, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988) has opined that it is "error to find obviousness where references 'diverge from and teach away from the invention at hand'." *Id.* As noted above, because Furber teaches the desirability of the at least two paths operating at equal speed and having any tie broke by the "priority" of the instruction, it clearly would lead one of ordinary skill in the art away from having one path operate "more quickly" than the other paths. The Examiner fails to explain how or why one would ignore this explicit teaching in Furber and, instead adopt the claimed "more quickly" feature.

As a result of the above, it is clear that Furber teaches away from the claimed invention.

G. Claims 1-6, 10-16, 20 and 21 are not obvious under 35 USC §103 over Furber/Patterson

Because many of the arguments in previous sections A-F apply to both stated grounds of rejection, Appellant, in order to save the duplication of arguments by incorporating by reference the enumerated section's discussion.

Sections A and B above establishes that the Examiner fails to support his contention that Furber teaches Appellant's claimed "determination" between

address generation paths based on an IFCI being the first prefetched instruction and fails to teach one path operating "more quickly" than any other paths. As noted in Section C, the Examiner fails to establish where the two features missing from Furber are taught in Patterson.

The Court of Appeals for the Federal Circuit has held that "the PTO has the burden under Section 103 to establish a *prima facie* case of obviousness." *In re Fine* at 1598. "It can satisfy this burden only by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the references." *Id.*

As noted in section D above, at least two features positively recited in Appellant's claims are not identified by the Examiner as being disclosed in either of the cited references. Without these features being specifically identified in at least one of the Furber or Patterson references, even if the references were combined in the manner suggested by the Examiner, the Examiner has failed to establish any *prima facie* case of obviousness and there would be no basis for rejection under §103.

As noted in section E above, the Federal Circuit, in order to avoid hindsight reasoning, requires the Examiner to enumerate some "reason" or "motivation" for the picking and choosing of elements from various references. Also as noted in section E above, the Examiner has failed to provide any reason or motivation to

combine the Furber and Patterson references and therefore has not met his burden of proving obviousness under §103.

Moreover, even **if** the cited references taught the features of the independent claims 1, 11 and 21 and **if** the Examiner had provided some reason or motivation for combining these references, the rejection would still fail because Furber teaches away from the claimed combination as discussed in section F above.

Accordingly, for any and all of the reasons noted above, the rejection under 35 USC §103 of claims 1, 11, and 21 over the Furber and Patterson combination is unsupported in the Final Rejection and should be reversed. In as much as claims 2-6 and 10 depend from claim 1 and claims 12-16 and 20 depend from claim 11 these dependent claims are non-obvious over the Furber/Patterson combination in view of the above arguments.

H. Claims 7-9 and 17-19 are not obvious under 35 USC §103 over the Furber/Patterson/Hara

Because many of the arguments in previous sections A-F apply to both stated grounds of rejection, Appellant, in order to save the duplication of arguments by incorporating by reference the enumerated section's discussion. Additionally, because this rejection of claims is based upon the Furber/Patterson

references and their combination, all of the above arguments in section G (and previous sections A-F) are herein incorporated by reference.

The Examiner makes no argument that the Hara reference teaches the two features missing from the Furber and Patterson references. The Examiner fails to identify any reason or motivation in the Hara reference for combining either Furber/Patterson or Furber/Patterson/Hara. The Examiner fails to indicate why one of ordinary skill would ignore the specific teachings in Furber away from the claimed invention and, instead, combine them with Hara to find the invention of claims 1, 11 and 21 or claims dependent thereon.

Accordingly, for any and all of the reasons noted above, the rejection under 35 USC §103 of claims 1, 11, and 21 over the Furber, Patterson and Hara combination is unsupported in the Final Rejection and should be reversed. In as much as claims 7-9 depend from claim 1 and claims 17-19 depend from claim 11 these dependent claims are non-obvious over the Furber/Patterson/Hara combination in view of the above arguments.

VIII. CONCLUSION

As discussed in detail above, the Furber reference does not teach address generation logic for use only when an IFCI is present and the determination as to use of a first address generation path is made based upon **if** a "selected prefetched instruction is a first prefetched instruction." Furber also fails to contain any

disclosure that there is any benefit to having the first address generation path generate a target address **more quickly** than another path. The Examiner has not indicated where the Patterson reference discloses either of the elements missing from Furber and, even if both were combined, they would still fail to disclose the subject matter of the claims. The Examiner has failed to provide any legal "reason" or "motivation" for combining the Furber and Patterson references and ignores the fact that Furber teaches away from the claimed combination. He therefore has simply failed to establish any *prima facie* case of obviousness. Claims 2-10 and 12-20 depend from independent claims 1 and 11, respectively, and therefore cannot be obvious in view of the Furber/Patterson combination. Claims 7-9 and 17-19 are rejected over the Furber/Patterson/Hara combination and Hara does not supply the missing elements nor the missing "reason" or "motivation" for combining references.

As a result of the above, there is simply no support for the rejections of Appellant's independent claims or claims dependent thereon under 35 USC §103. Thus, and in view of the above, the rejection of claims 1-21under 35 USC §103 is clearly in error and reversal thereof by this Honorable Board is respectfully requested.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: _____

Stanley C. Spooner
Reg. No. 27,393

SCS:kmm
Enclosure

IX. CLAIMS APPENDIX

1. A data processing apparatus, comprising:

a processor operable to execute a stream of instructions;

a prefetch unit operable to prefetch instructions from a memory prior to sending those instructions to the processor for execution, the prefetch unit being operable to receive from the memory simultaneously a plurality of prefetched instructions from sequential addresses in memory, and being operable to detect whether any of those prefetched instructions are an instruction flow changing instruction, and based thereon to output a fetch address for a next instruction to be prefetched by the prefetch unit;

address generation logic, within the prefetch unit and responsive to a selected prefetched instruction that is detected to be said instruction flow changing instruction, for determining a target address to be output as the fetch address, the address generation logic having a first address generation path for determining the target address if the selected prefetched instruction is a first prefetched instruction in said plurality, and at least one further address generation path for determining the target address if the selected prefetched instruction is one of the other prefetched instructions in said plurality, the first prefetched instruction being earlier in said stream than said other prefetched instructions, the first address generation path generating the target address more quickly than the at least one further address generation path; and

a pipeline stage, provided in said at least one further address generation path, for increasing generation speed of the target address by the first address generation path, whereby, in the event that the first prefetched instruction is said selected prefetched instruction, the prefetch unit outputs the associated target address as the fetch address earlier than if one of said other prefetched instructions is said selected prefetched instruction.

2. A data processing apparatus as claimed in Claim 1, further comprising:

prediction logic operable to predict, for a prefetched instruction whose execution is conditional, whether that conditional prefetched instruction will be executed by the processor;

in the event that the first prefetched instruction is said selected prefetched instruction and its execution is conditional, the prefetch unit being operable to output the associated target address as the fetch address if the prediction logic predicts that the first prefetched instruction will be executed by the processor.

3. A data processing apparatus as claimed in Claim 1, wherein the prefetch unit associates a different priority level with each of the plurality of prefetched instructions, the first prefetched instruction having the highest priority level, and if more than one of the plurality of prefetched instructions is detected to be said instruction flow changing instruction, the prefetch unit is operable to determine as

said selected prefetched instruction the prefetched instruction having the higher priority level from those prefetched instructions detected to be said instruction flow changing instruction, whereby the target address associated with that selected prefetched instruction is output as the fetch address.

4. A data processing apparatus as claimed in Claim 1, wherein the address generation logic is operable to generate the target address for the first prefetched instruction in a same clock cycle as the prefetch unit detects that that first prefetched instruction is said instruction flow changing instruction.

5. A data processing apparatus as claimed in Claim 1, wherein predetermined logic is shared between the first address generation path and the at least one further address generation path.

6. A data processing apparatus as claimed in Claim 1, wherein the at least one further address generation path comprises a single further address generation path used to determine the target address for any prefetched instructions other than said first prefetched instruction.

7. A data processing apparatus as claimed in Claim 1, wherein the prefetch unit comprises decode logic operable to detect whether any of the plurality of

prefetched instructions are said instruction flow changing instruction, the decode logic further being operable to decode from each prefetched instruction detected to be said instruction flow changing instruction an immediate value to be input to the address generation logic.

8. A data processing apparatus as claimed in Claim 7, wherein the address generation logic comprises adder logic operable to determine the target address for the selected prefetched instruction by adding the associated input immediate value to the address of that selected prefetched instruction.

9. A data processing apparatus as claimed in Claim 8, wherein the adder logic is shared between the first address generation path and the at least one further address generation path.

10. A data processing apparatus as claimed in Claim 1, wherein if none of the plurality of prefetched instructions is said instruction flow changing instruction, the prefetch unit is operable to generate the fetch address by incrementing a previous fetch address output by the prefetch unit.

11. A method of operating a data processing apparatus to determine a target address for an instruction flow changing instruction, the data processing apparatus

having a processor operable to execute a stream of instructions, and a prefetch unit operable to prefetch instructions from a memory prior to sending those instructions to the processor for execution, and to output a fetch address for a next instruction to be prefetched from the memory, the method comprising the steps of:

(a) receiving from the memory simultaneously a plurality of prefetched instructions from sequential addresses in memory;

(b) detecting whether any of those prefetched instructions are an instruction flow changing instruction; and

(c) for a selected prefetched instruction that is detected to be said instruction flow changing instruction, determining a target address to be output as the fetch address by performing one of the steps of:

(c)(1) employing a first address generation path to determine the target address if the selected prefetched instruction is a first prefetched instruction in said plurality; or

(c)(2) employing at least one further address generation path to determine the target address if the selected prefetched instruction is one of the other prefetched instructions in said plurality;

the first prefetched instruction being earlier in said stream than said other prefetched instructions, and the first address generation path being arranged to generate the target address more quickly than the at least one further address generation path;

(d) providing a pipeline stage in the at least one further address generation path in order to increase speed of generation of the target address by the first address generation path; and

(e) outputting as the fetch address the target address generated at step (c);

whereby, in the event that the first prefetched instruction is said selected prefetched instruction, the prefetch unit is operable to output the associated target address as the fetch address earlier than if one of said other prefetched instructions is said selected prefetched instruction.

12. A method as claimed in Claim 11, further comprising the step of:

employing prediction logic to predict, for a prefetched instruction whose execution is conditional, whether that conditional prefetched instruction will be executed by the processor;

in the event that the first prefetched instruction is said selected prefetched instruction and its execution is conditional, the prefetch unit being operable to output the associated target address as the fetch address if the prediction logic predicts that the first prefetched instruction will be executed by the processor.

13. (original) A method as claimed in Claim 11, further comprising the steps of:

associating a different priority level with each of the plurality of prefetched instructions, the first prefetched instruction having the highest priority level;

if more than one of the plurality of prefetched instructions is detected to be said instruction flow changing instruction, determining as said selected prefetched instruction for said step (c) the prefetched instruction having the higher priority level from those prefetched instructions detected to be said instruction flow changing instruction;

whereby at said step (d) the target address associated with that selected prefetched instruction is output as the fetch address.

14. A method as claimed in Claim 11, wherein at said step (c)(1) the target address for the first prefetched instruction is generated in a same clock cycle that, during said step (b), that first prefetched instruction is detected as said instruction flow changing instruction.

15. A method as claimed in Claim 11, wherein predetermined logic is shared between the first address generation path and the at least one further address generation path.

16. A method as claimed in Claim 11, wherein the at least one further address generation path comprises a single further address generation path used at said step

(c)(2) to determine the target address for any prefetched instructions other than said first prefetched instruction.

17. A method as claimed in Claim 11, wherein the prefetch unit comprises decode logic operable at said step (b) to detect whether any of the plurality of prefetched instructions are said instruction flow changing instruction, the method further comprising the step of:

employing the decode logic to decode from each prefetched instruction detected to be said instruction flow changing instruction an immediate value for use in said step (c).

18. A method as claimed in Claim 17, wherein at said step (c) the target address for the selected prefetched instruction is determined by adding the associated immediate value to the address of that selected prefetched instruction.

19. A method as claimed in Claim 18, wherein adder logic used to perform said adding step is shared between the first address generation path and the at least one further address generation path.

20. A method as claimed in Claim 11, wherein if none of the plurality of prefetched instructions is determined at said step (b) to be said instruction flow changing instruction, the method further comprises the step of:

generating the fetch address by incrementing a previous fetch address output by the prefetch unit, and outputting that fetch address at said step (d).

21. A prefetch unit for a data processing apparatus that has a processor operable to execute a stream of instructions, the prefetch unit being operable to prefetch instructions from a memory prior to sending those instructions to the processor for execution, the prefetch unit being operable to receive from the memory simultaneously a plurality of prefetched instructions from sequential addresses in memory, and being operable to detect whether any of those prefetched instructions are an instruction flow changing instruction, and based thereon to output a fetch address for a next instruction to be prefetched by the prefetch unit, the prefetch unit comprising:

address generation logic, responsive to a selected prefetched instruction that is detected to be said instruction flow changing instruction, for determining a target address to be output as the fetch address, the address generation logic having a first address generation path for determining the target address if the selected prefetched instruction is a first prefetched instruction in said plurality, and at least one further address generation path for determining the target address if

the selected prefetched instruction is one of the other prefetched instructions in said plurality, the first prefetched instruction being earlier in said stream than said other prefetched instructions, the first address generation path generating the target address more quickly than the at least one further address generation path; and

a pipeline stage, provided in said at least one further address generation path, for increasing generation speed of the target address by the first address generation path, whereby, in the event that the first prefetched instruction is said selected prefetched instruction, the prefetch unit outputs the associated target address as the fetch address earlier than if one of said other prefetched instructions is said selected prefetched instruction.

X. EVIDENCE APPENDIX

None.

XI. RELATED PROCEEDINGS APPENDIX

None.